SANDIA REPORT

SAND2014-19336
Unlimited Release
Printed October 2014

Spherical expansion of a dense plasma: test of ambipolar field model in Aleph

Thomas P. Hughes and Lawrence C. Musson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: http://www.ntis.gov/search

SAND2014-xxxx
Unlimited Release
Printed May 2014

Spherical expansion of a dense plasma: test of ambipolar field model in Aleph

Thomas P. Hughes
Applied Science and Technology Maturation
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-MS0878
thughe@sandia.gov

Lawrence C. Musson
Electrical Models and Simulation
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-MS1177
lcmusso@sandia.gov

Abstract

The ambipolar field model in Aleph replaces kinetic electrons with an ideal gas whose density is equal to the local ion density and whose temperature is a specified uniform value. This approximation allows one to avoid the numerical constraints on timestep and spatial resolution imposed by the plasma frequency and Debye length, respectively. We derive an analytic model for ambipolar spherical expansion and compare it to the results of Aleph simulations. The ion density and average ion energy as functions of radius agree with the analytic model. However, the algorithm generates significant numerical heating of the ions. This can be reduced, to an extent, by decreasing the timestep and using more macroparticles to represent the ions.

Acknowledgments

We thank Alan Scott (ParaView support) for his help with post-processing and visualizing the numerical results.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

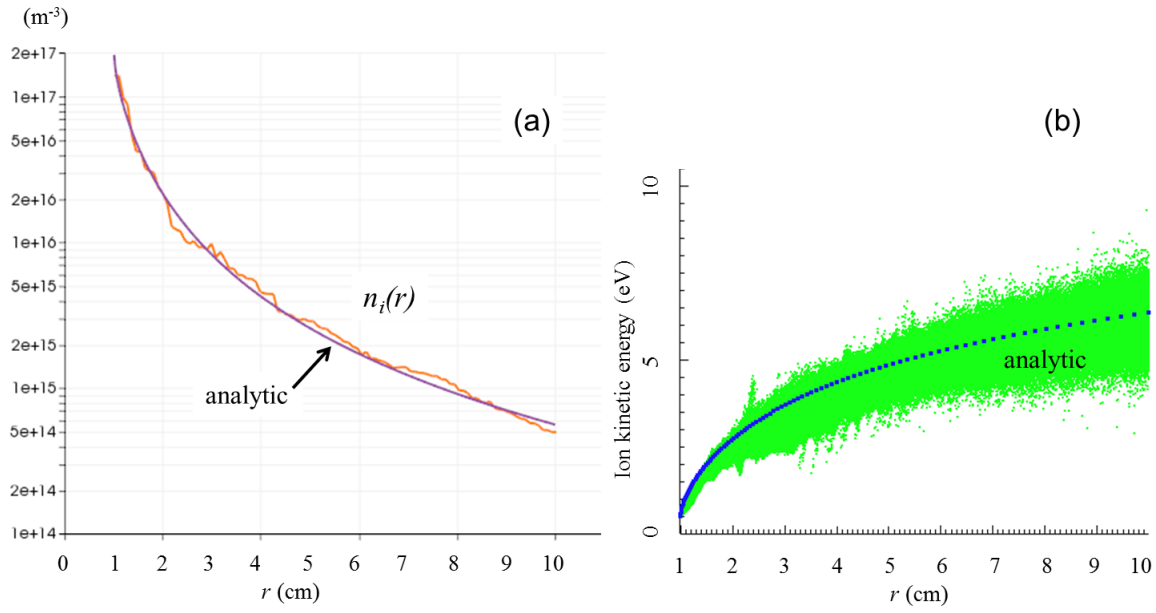

Figure 1. Spherical expansion of an isothermal plasma with $T_e = 1$ eV. Plot (a) compares radial ion density profile from Aleph (version 4454) to analytic model and plot (b) compares ion energy vs. radius to analytic model. See Sec. 4.2 for details of this calculation.

Executive Summary

The ability to treat plasma expansion over a large density range is needed to follow the motion of dense plasmas generated at electrode and insulator surfaces in vacuum. Using an explicit, kinetic, particle-in-cell model often results in prohibitively long calculations due to numerical stability constraints imposed by light, fast-moving electrons. As a result, one would like to use models where these constraints can be avoided. One of the simplest models is to replace the kinetic electrons with an ideal electron gas characterized only by a density and temperature. The temperature has a user-specified value and the density is obtained from the quasineutrality assumption, namely, that the electron gas density is locally equal to the ion density. The ions are advanced using a standard kinetic, particle-in-cell algorithm. This treatment is called the "ambipolar" algorithm in Aleph. In this report, we evaluate the algorithm by using it to model a spherically-expanding plasma where the density changes by about a factor of 360. We derive an analytic solution to this problem and use it to check the Aleph results. The radial density profile agrees with the analytic model, as shown in Fig. 1(a). We observe that the algorithm produces significant numerical heating of the ions (Fig. 1(b)). This because the ambipolar electric field is obtained from the gradient of the ion density. Reducing the timestep and increasing the number of macroparticles per cell both reduce the numerical heating, at the cost of increased computation time.

Contents

Appendix

1 Introduction

Aleph's ambipolar-field algorithm treats a dense plasma by advancing kinetic ions in the electric field generated by the thermal pressure of the neutralizing electrons. This algorithm allows one to go to much higher plasma densities than is practical using an explicit kinetic algorithm for the electrons. Various implicit treatments of kinetic-electron dynamics have been developed over the years [1–3], but are not currently available in Aleph. To evaluate the ambipolar algorithm, we use it to model spherical expansion of a plasma. We assume the existence of a steady-state spherically-symmetric plasma source inside a small region. Quasineutral plasma flows out from this region, and expands to a radius ten times that of the source region boundary. In the simulation we do not model the source, only the plasma expansion between spherical inflow and outflow surfaces. The simulation volume is shown in Fig. 2.

In Sec. 2, we give the equations for the ambipolar algorithm and apply a normalization scheme to identify the free parameters. In Sec. 3 we derive an analytic solution for steady-state spherically-symmetric expansion, and show that the ions have to be injected at or above the ion "sound speed" at the inflow surface. We give analytic results for density, velocity, and electric field as functions of radius. Section 4 describes the Aleph calculations, and compares the results to the analytic model. We carried out a "reference" simulation and then varied several numerical parameters to see their effect on the electric field noise level. A discussion of the results is given in Sec. 5.

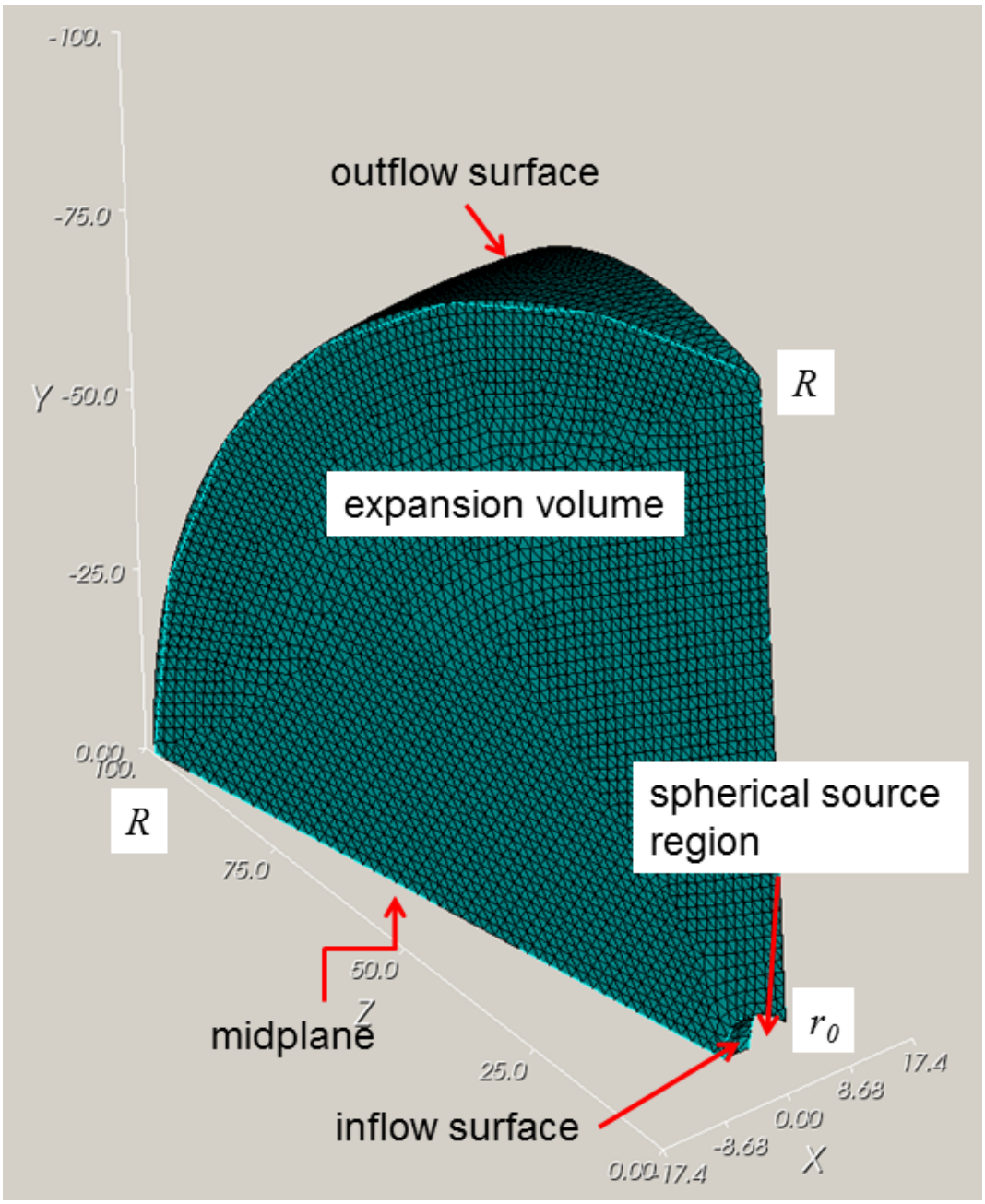

Figure 2. Model geometry for plasma expansion and tetrahedral finite-element mesh created by Cubit. Axis units are mm. Cubit 14.1 input file is in Appendix A.

2 Equations for Aleph ambipolar model

In Aleph, the equation of motion of an ion is

$$m_i \frac{d\mathbf{v}_i}{dt} = Z_i e \mathbf{E}(\mathbf{x}_i), \tag{1}$$

where m_i is the ion mass, $\mathbf{v}_i$ is the non-relativistic velocity of the ion at position $\mathbf{x}_i$, Z_i is the integer charge-state of the ion, and $\mathbf{E}$ is the electric field at $\mathbf{x}_i$. Rather than solving Poisson's equation for the electric field $\mathbf{E}(\mathbf{x})$ (which is precluded by the quasineutrality assumption), we obtain $\mathbf{E}$ by assuming that the electron gas is always approximately in force balance:

$$n_e m_e \frac{d\mathbf{v}_e}{dt} \approx 0 \approx -n_e e \mathbf{E} - \nabla p_e, \tag{2}$$

where n_e is the electron density, $\mathbf{v}_e$ is the electron mean velocity, and p_e is the electron pressure. Aleph uses the isothermal ideal-gas relation $p_e = n_e k T_e$ for the electrons, where T_e is a specified constant, uniform, electron temperature. Equation 2 gives

$$\mathbf{E} \approx -\frac{kT_e}{en_e}\nabla n_e \approx -\frac{kT_e}{e}\frac{\nabla n_i}{n_i}, \tag{3}$$

where we have used quasineutrality to replace n_e with the ion density n_i. Equation 1 now becomes

$$\frac{d\mathbf{v}_i}{dt} = -\frac{Z_i k T_e}{m_i n_i}\nabla n_i = -c_i^2 \frac{\nabla n_i}{n_i}, \tag{4}$$

where $c_i \equiv \sqrt{Z_i k T_e / m_i}$ is the ion sound speed (we assume $T_i \ll T_e$). For a spherically-expanding plasma, the right side of Eq. 4 is positive since $\nabla n_i < 0$, and therefore the ions are accelerated outward by the ambipolar electric field. Equation 4 is the equation used in Aleph to advance the ions. The ambipolar electric field model is invoked by the line

```
electromagnetic model for block_1 = ambipolar, Te = {Te_K}, ...
```

in the Aleph (version 4554) input file (see Appendix B).

Table 1. Normalization of variables.

Quantity:	Symbol	Normalize to:
Radial position	r	R
Time	t	R/c_i
Ion radial velocity	v	c_i
Ion kinetic energy	$\frac{1}{2}m_i v^2$	mc_i^2
Ion density	n	n_0
Electric potential	ϕ	kT_e/e
Radial electric field	E	kT_e/Re
Ion potential energy	$Z_i e\phi$	$Z_i kT_e$

3 Analytic solution for steady-state expansion

We can obtain an analytic solution for a steady-state spherically expanding plasma obeying Eq. 4. To simplify the equations, we use the normalization scheme in Table 1, where R is the radius of the outflow surface in Fig. 2, and n_0 is the ion density at the inflow radius r_0. Using caret symbols to denote normalized quantities, the radial component of Eq. 4 becomes

$$\frac{d\hat{v}}{d\hat{t}} = -\frac{\hat{\nabla}\hat{n}}{\hat{n}}. \tag{5}$$

Using the chain rule

$$\frac{d}{d\hat{t}} = \frac{d\hat{r}}{d\hat{t}}\frac{d}{d\hat{r}} = \hat{v}\frac{d}{d\hat{r}}, \tag{6}$$

Eq. 5 becomes

$$\hat{v}\frac{d\hat{v}}{d\hat{r}} = -\frac{1}{\hat{n}}\frac{d\hat{n}}{d\hat{r}}, \tag{7}$$

which we can integrate to obtain

$$\frac{\hat{v}^2}{2} = \frac{\hat{v}_0^2}{2} - \log\hat{n}, \tag{8}$$

where $\hat{v}_0$ is the normalized ion velocity at the inflow surface. This equation describes the conservation of energy, in normalized form, for the ions. In steady state, the total ion flux is independent of radius, so we obtain (since $\hat{n} = \hat{n}_0 = 1$ at $\hat{r}_0$)

$$\hat{n}\hat{v}\hat{r}^2 = \hat{v}_0\hat{r}_0^2. \tag{9}$$

Substituting for $\hat{n}$ in Eq. 8 and rearranging terms gives

$$\frac{\hat{r}}{\hat{r}_0} = \sqrt{\frac{\hat{v}_0}{\hat{v}}}\exp\left(\frac{\hat{v}^2 - \hat{v}_0^2}{4}\right). \tag{10}$$

This is an implicit equation for $\hat{v}(\hat{r})$. We can show that it imposes a constraint on the ion inflow velocity $\hat{v}_0$. Expanding the RHS in $\delta\hat{v} = \hat{v} - \hat{v}_0$ gives

$$\frac{\hat{r}}{\hat{r}_0} \approx 1 + \frac{\delta\hat{v}}{2\hat{v}_0}\left(\hat{v}_0^2 - 1\right). \tag{11}$$

We see that in order for an ion to accelerate ($\delta\hat{v} > 0$) rather than decelerate for $\hat{r} > \hat{r}_0$, we require $\hat{v}_0 > 1$. In physical units, this means that the ions must be injected with a supersonic velocity $v_0 > c_i$. This is analogous to the Bohm condition for ions entering a non-neutral sheath at a planar wall/plasma interface. The kinetic energy of an ion with velocity c_i is $\frac{1}{2}m_i c_i^2 = \frac{1}{2}Z_i k T_e$. Thus, if the electron temperature is 1 eV, the ions must be injected with a minimum kinetic energy of 0.5 eV (for $Z_i = 1$).

The free parameters in this system are the electron temperature T_e, the inflow-boundary parameters r_0, n_0, v_0 (subject to the above constraint), and the outer radius R. Radial profiles of all the physical quantities can be obtained from a set of universal curves parametrized by the value $\hat{v}_0 = v_0/c_i$. Figures 3 and 4 give ion density and kinetic-energy profiles for $\hat{v}_0 = 1$ (the minimum value) and $\hat{v}_0 = \sqrt{5}$ (a factor of 5 larger initial kinetic energy). For $\hat{\nu}_0 = 1$, we see that the ion density drops by a factor of about 360, and the ions accelerate to about 13 times their initial energy (0.5 eV to 6.4 eV) as the plasma expands by a factor of 10 in radius. At five times the minimum energy, these factors drop to about 180 (density) and 3 (energy).

Two further useful expressions can be derived. The radial electric field is given by

$$\hat{E}_r(\hat{r}) = \frac{2}{\hat{r}}\left(\frac{\hat{v}^2}{\hat{v}^2 - 1}\right) \to \frac{2}{\hat{r}} \quad \text{for large } \hat{v},\, \hat{r}, \tag{12}$$

and the time for an ion to travel from $\hat{r}_0$ to a given radius $\hat{r} > \hat{r}_0$ is

$$\hat{t} = \int_{\hat{r}_0}^{\hat{r}} \frac{d\hat{r}}{\hat{v}(\hat{r})} = \frac{\sqrt{\hat{v}_0}r_0}{2}\int_{\hat{v}_0}^{\hat{v}(\hat{r})} \frac{\hat{v}^2 - 1}{\hat{v}^{5/2}} \exp\left(\frac{\hat{v}^2 - \hat{v}_0^2}{4}\right) d\hat{v}. \tag{13}$$

A plot of the radial electric field from Eq. 12 is shown in Fig. 5. For the minimum value of inflow ion energy, the electric field is singular at r_0. The singularity is integrable, so that the energy acquired by an ion passing through it is finite (see Fig. 4). From this figure and from Eq. 12, we also see that at large radius, the electric field is independent of v_0.

Equation 13 can be integrated numerically to find the time it takes for an ion to travel a given distance. For example, an ion starting at the minimum energy takes $t = 0.31R/c_i$ to travel from r_0 to $R = 10r_0$.

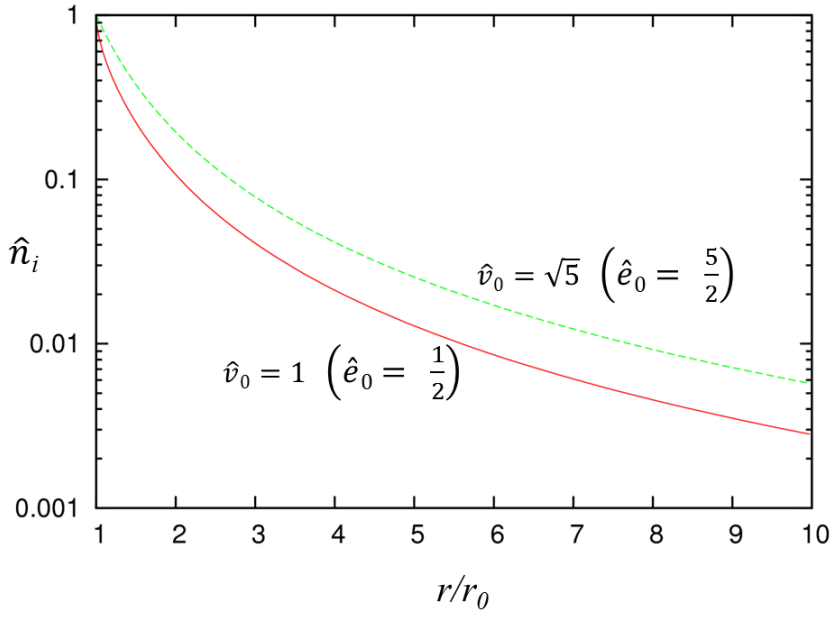

Figure 3. Normalized ion density as a function of radius in steady state for two values of normalized ion inflow velocity (corresponding normalized initial kinetic energies are shown in parentheses).

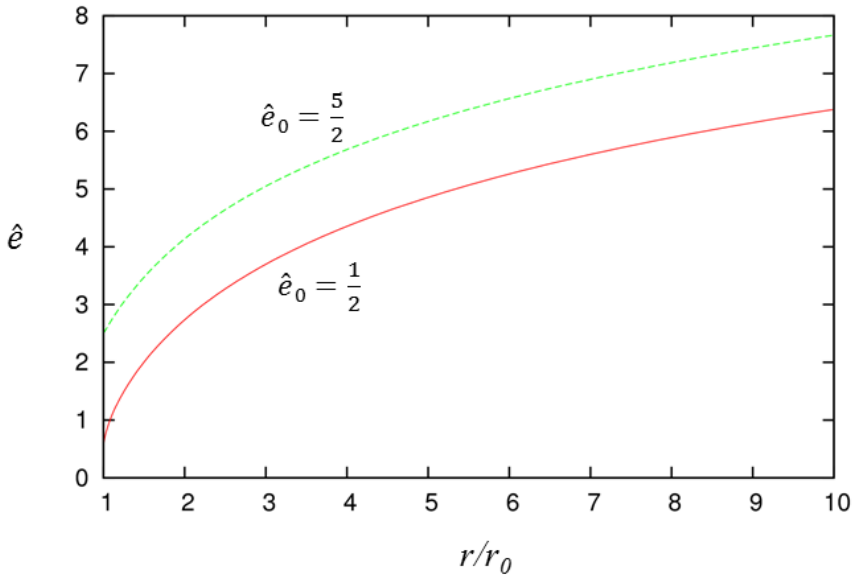

Figure 4. Normalized ion energy as a function of radius in steady state, for two values of normalized ion inflow kinetic energy. Energy is normalized to $Z_i kT_e$.

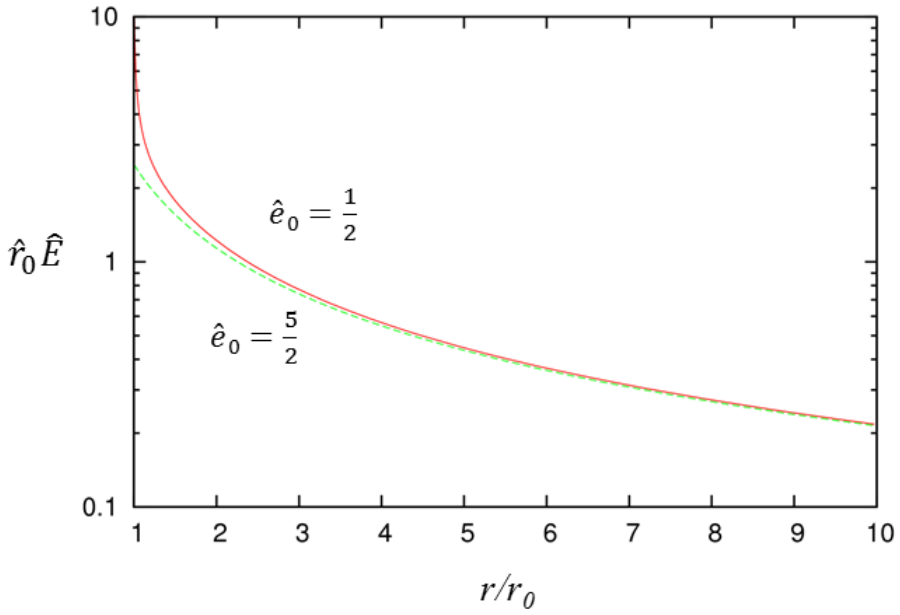

Figure 5. Normalized radial electric field vs. radius for two values of the normalized ion inflow kinetic energy.

Table 2. Physical parameters for Aleph simulations. For normalization, see Table 1.

Parameter	Symbol	Value	Unit	Normalized value
Inner radius	r_0	0.01	m	0.1
Outer radius	R	0.1	m	1
Injected plasma density	n_0	10^{16}	m^{-3}	1
Electron temperature	kT_e	1	eV	
Ion mass	m_i	1.66×10^{-27}	kg	
Ion charge state	Z_i	1	–	
Ion sound speed	c_i	9.8×10^3	m/s	1
Ion inflow speed	v_0	9.8×10^3	m/s	1
Time limit	T	1.8×10^{-5}	s	1.76

4 Aleph results

The geometry for the Aleph calculations, shown in Fig. 2, consists of one half of a hollow, spherical, 20° wedge. This shape is used to obtain the smallest mesh consistent with spherical symmetry in a 3D finite-element code. (Aleph does not have a spherical– or cylindrical–coordinates option.) The plasma inflow and outflow surfaces are spherical, with radii of 10 and 100 mm, respectively. The remaining sides of the expansion volume are planar surfaces where mirror-symmetry boundary conditions are imposed. Ions are injected uniformly and continuously at the inner radius ($r = r_0$) of the meshed region at the ion sound speed, which, as we saw in Sec. 3, is the smallest velocity consistent with a steady-state solution. The presence of neutralizing electrons is accounted for through the electric field (see Sec. 2). The ions expand into the vacuum region and are absorbed when they reach the outflow surface.

The physical simulation parameters are given in Table 2, and this set of values was used in all the simulations. Numerical parameters are given in Table 3. Square brackets contain lists of values used in different simulations. The "macroparticle weight", W_p, is the number of real ions represented by one simulation macroparticle. The value is obtained by first picking a desired value of the "macroparticles/cell" parameter near the inflow surface, and then calculating the macroparticle weight needed to get this number given the plasma density and cell volume. We used the expression $\Delta x^3/6\sqrt{2}$ for the volume of a regular tetrahedron with edge length Δx (although the tetrahedrons generated by Cubit are not regular, in general). Smaller values of W_p mean that more macroparticles are needed to represent a given physical density, and this gives better resolution of the particle distribution.

Temporal averaging is used in Aleph to reduce numerical noise. In these calculations, we used the moving-window option, where the code keeps the last N_{win} values of the density at each node, and averages these on each timestep. The ambipolar field is then calculated from Eq. 3 using these averaged density values.

Table 3. Numerical parameters for Aleph simulations. Where more than one value is given, the first value is the one used in the "reference" simulation. For normalization, see Table 1.

Parameter	Symbol	Value	Unit	Normalized value
Cubit mesh size	Δx	1.745	mm	0.01745
Timestep	Δt	[60, 6]	ns	5.87×10^{-3}
Macroparticles/cell at r_0	N_c	[100, 1000]	ions	
Macroparticle weight	W_p	[12, 1.2]$\times 10^5$	ions	
Smoothing window	N_{win}	[100, 1000, 1]	steps	

4.1 Reference-simulation results

Our "reference" simulation is the one using the first values in the square brackets in Table 3. Results are shown in Figs. 6–8.

The 2D color-map plots in Fig. 6 show the ion density and E_z field component in the horizontal x–z "mid-plane" of the spherical wedge (see Fig. 2). In this plane, the z–axis is the equivalent of the radial coordinate in spherical coordinates. The radial density profile in Fig. 6(c) is reasonably smooth and agrees well with the analytic result. The radial electric field $E_z(z)$ in Fig. 6(d) agrees on average with the analytic result, but is very noisy. The noise is due to the fact that in the ambipolar approximation, **E** is obtained from the spatial gradient of the density (Eq. 3), and this amplifies the particle noise already in the density. The noise in **E** leads to a large spread in the ion energy, as shown in Fig. 7. The average ion acceleration in Fig. 7(b) follows the analytic result. The energy distribution for the group of ions near $r = 5\,\mathrm{cm}$ in Fig. 8(a) is plotted in Fig. 8(b), and shows the large noise-induced spread about the analytic value. (The height of the vertical analytic line in Fig. 8(b) and similar figures below is arbitrary.)

4.2 Noise-reduction calculations

We carried out the following calculations to attempt to reduce the electric-field noise seen in the reference simulation:

1. Reduce timestep by a factor of 10, with several values of the temporal-smoothing window size (N_{win}).
2. Increase number of macroparticles by a factor of 10.
3. Combine (1) and (2).

These measures reduce the noise, as described below, though considerable noise still remains. Using factors larger than 10, one may be able to reduce the noise to any desired level, at

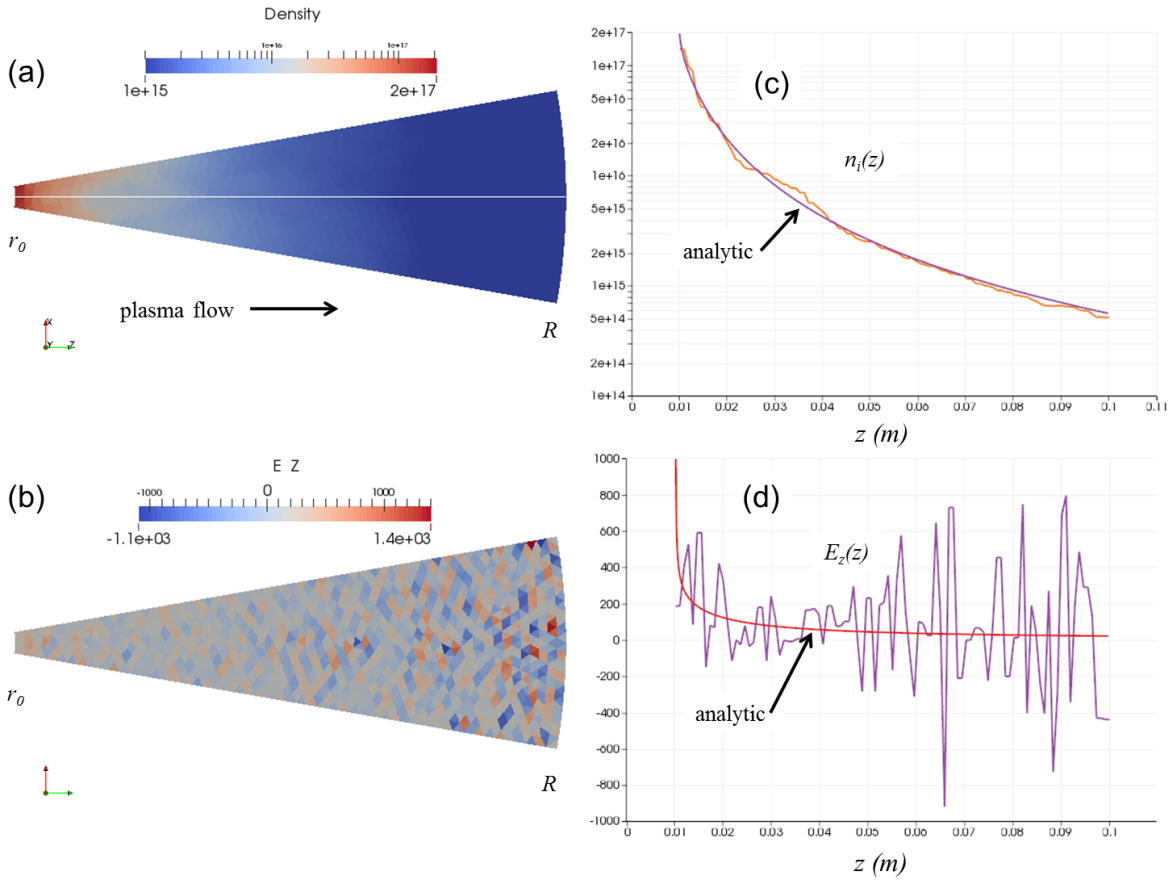

Figure 6. Field plots from Aleph reference simulation at 18 μs (MKS units): (a) shows density color map, and (b) shows the E_z field component color map. Both of these are in the x–z mid-plane (Fig. 2). Plot (c) compares the radial density profile $n_i(z)$ with the analytic result, and (d) compares $E_z(z)$ with the analytic result. Plots (c) and (d) are along the radial white line in (a).

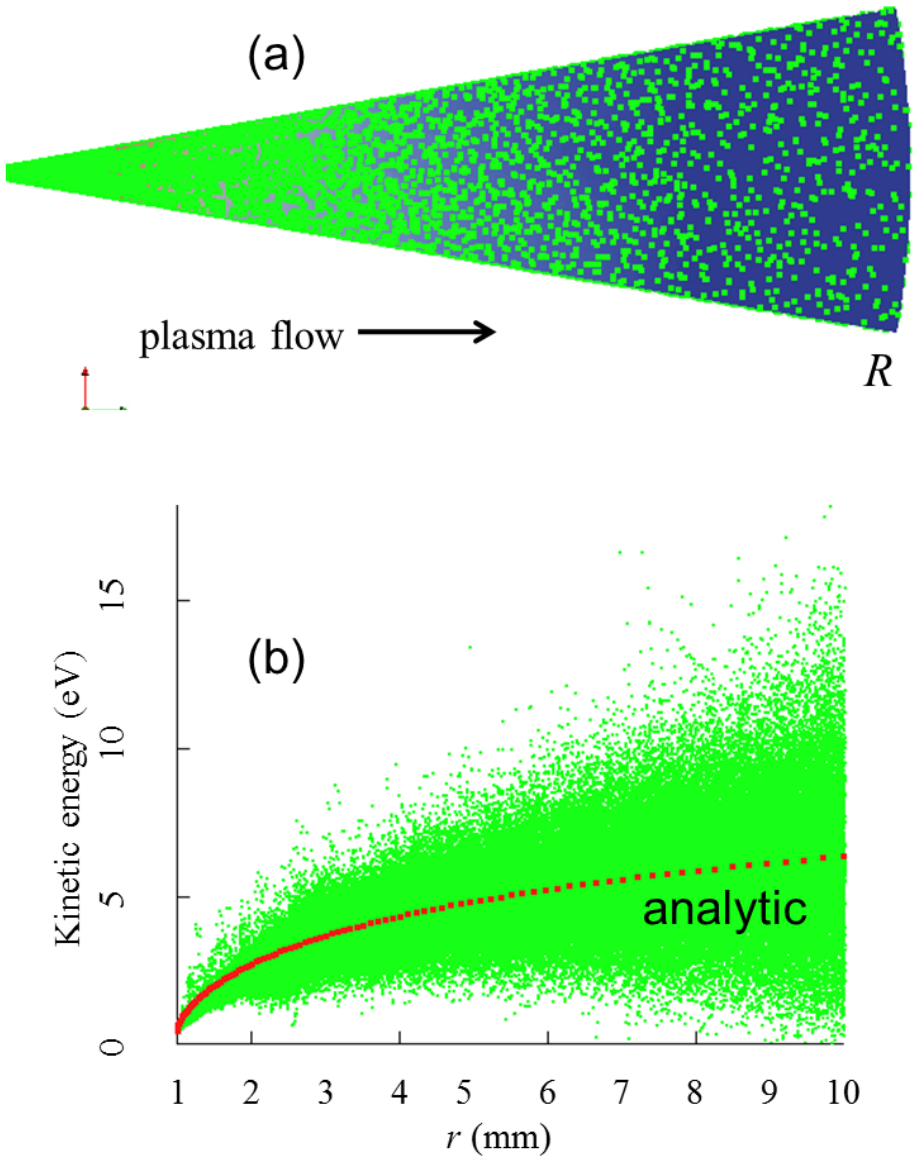

Figure 7. Particle plots from Aleph reference simulation at 18 μs showing (a) spatial positions of ions, and (b) comparing ion energies vs. radius to the analytic result.

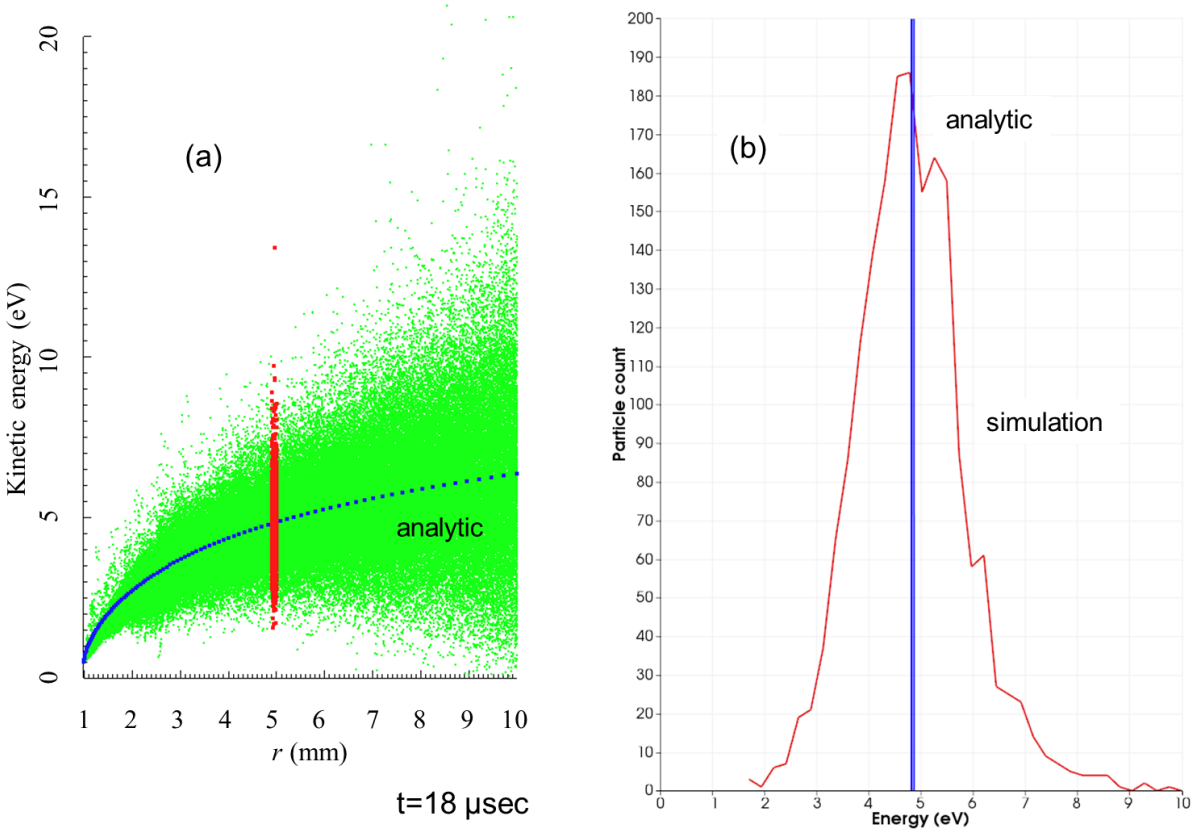

Figure 8. Ion energy vs. radius for the reference simulation is shown in (a). Energy distribution for red-colored ion sample near $r = 5\,\mathrm{cm}$ is plotted in (b).

the expense of increased computation time. Spatial smoothing was not tried, but may be beneficial. The physical parameters (Table 2) were kept the same for all runs.

Reduced timestep

For the timestep $\Delta t = 60$ ns used in the reference simulation, ions with the minimum injection velocity move approximately $\Delta x/2$ in one timestep, where Δx is the cell edge length. On each timestep, the ion density at each node is calculated from the ions that lie within one cell of the node. Reducing the timestep is motivated by the idea that this will reduce the density fluctuations from one timestep to the next. We used $\Delta t' = 6$ ns in this calculation, a factor of 10 reduction from the reference simulation. The width of the moving-average window, $t_{win} = N_{win}\Delta t$, was kept the same (6 μsec) by increasing N_{win} from 100 to 1000. No other numerical parameters were changed. The results in Fig. 9 show a significant reduction in the particle energy spread.

Additional calculations with $N_{win} = 100$ and N_{win}=1 gave results very similar to those in Fig. 9. This indicates that the reduced noise comes from reducing the timestep, and that temporal smoothing has little effect on the noise.

In Fig. 9(b), we see that the mean simulated ion energy at $r \approx 5$ cm is less than the analytic value. This is mainly a result of the singularity in the electric field at r_0 when the ions are injected at the minimum velocity v_0 (Fig. 5). Finer mesh resolution near the injection boundary should improve agreement in the mean energy. Injecting the ions at a larger initial velocity would eliminate the singularity in **E**.

Increased number of macroparticles

Here we increased the number of macroparticles by a factor of 10. In Aleph, this is achieved by reducing the macroparticle "weight" so that more macroparticles are used to represent the physical density. Other parameters were the same as in the reference simulation. The results in Fig. 10 again show a reduction of the noise level, though not as much as by reducing the timestep. The extra particles reduce the noise near the outer radius, where the particles/cell count in the reference simulation drops below 1.

Reduced timestep and increased macroparticles

This calculation combines the factors-of-ten reduction in timestep and increase in particle count used in the previous two calculations. The results in Fig. 11 have the lowest noise of the three modified calculations, indicating that the noise-reductions obtained with a smaller timestep and by adding more macroparticles appear to be additive.

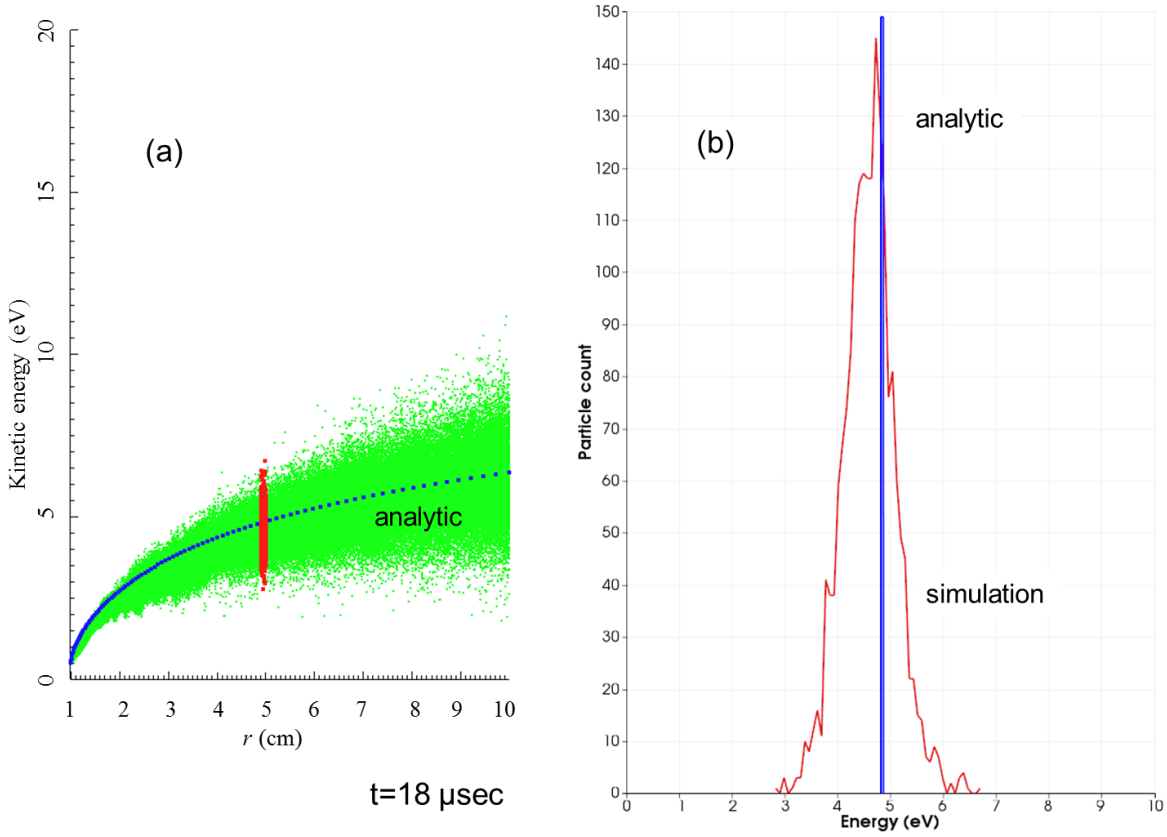

Figure 9. Ion energy vs. radius for simulation with timestep reduced by factor of 10 is shown in (a), along with the analytic result. Energy distribution for red-colored ion sample near $r = 5\,\mathrm{cm}$ is plotted in (b).

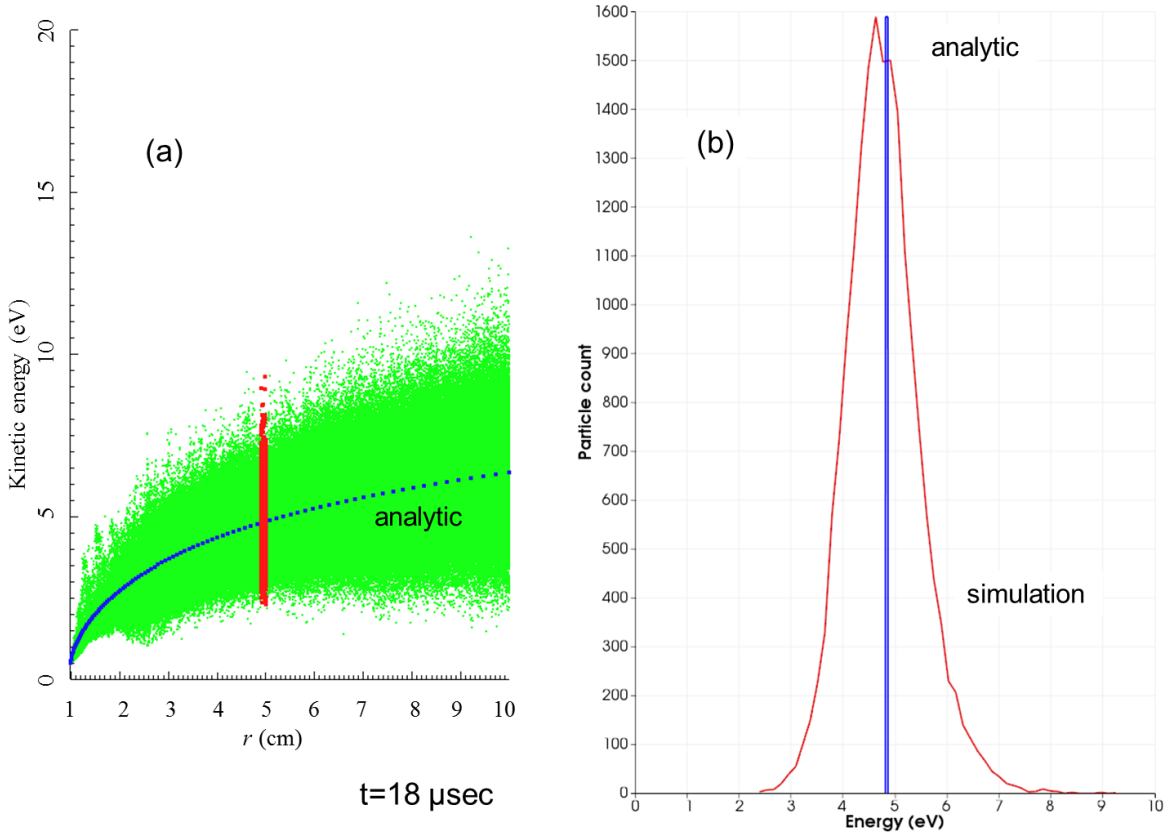

Figure 10. Ion energy vs. radius for simulation with factor of 10 more macroparticles is shown in (a), along with analytic result. Energy distribution for red-colored ion sample near $r = 5\,\mathrm{cm}$ is plotted in (b).

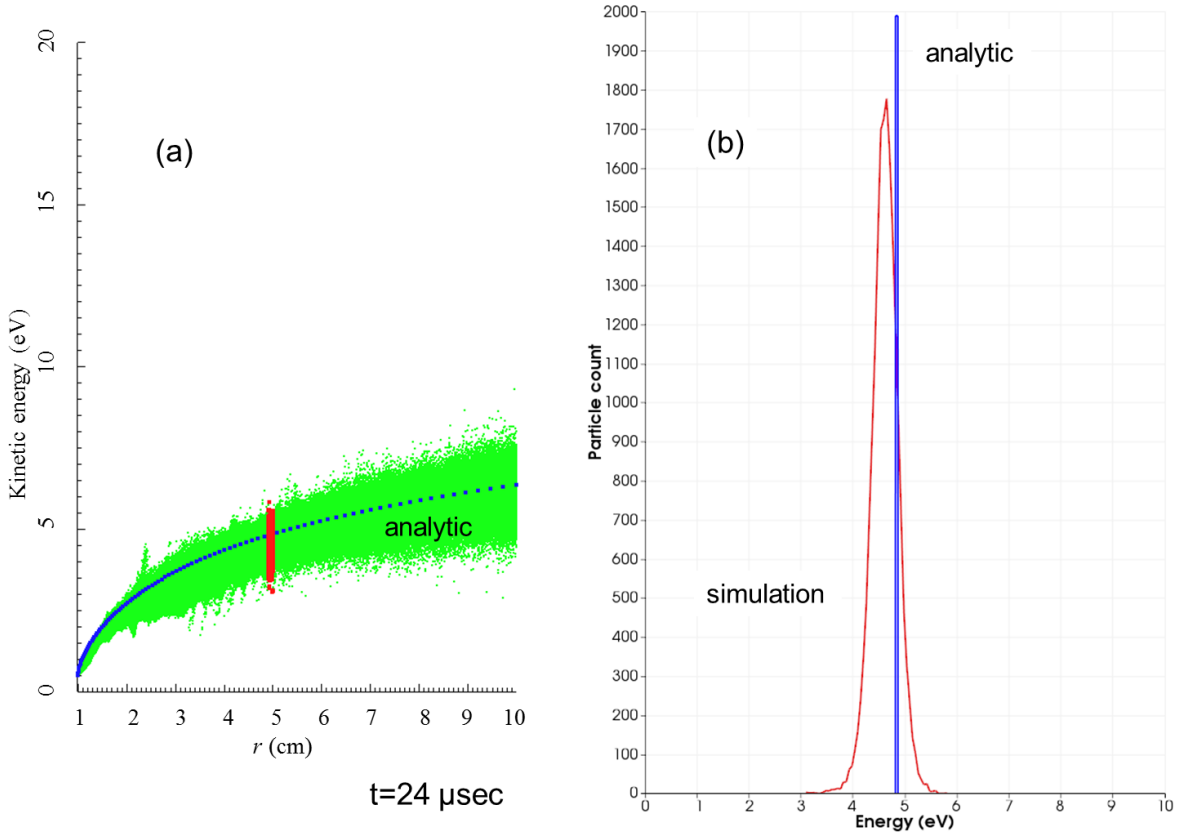

Figure 11. Ion energy vs. radius for simulation with both reduced timestep and more macroparticles is shown in (a), along with the analytic result. Energy distribution for red-colored ion sample near $r = 5\,\mathrm{cm}$ is plotted in (b).

5 Discussion

These results show that the ambipolar algorithm in Aleph is stable and robust. Because the electric field is computed from the derivative of the ion density, it amplifies the noise in the ion density. The resulting electric field noise leads to numerical heating of the ions. By contrast, the usual method of calculating the electric field from Poisson's equations has a smoothing effect on numerical noise in the particle density.

An analytic solution for steady-state isothermal spherical expansion from a source region was derived. The solution gives the radial profiles of ion density and energy. A minimum initial ion kinetic energy equal to one-half of the electron thermal energy is required for steady-state expansion. Ions starting with the minimum energy are accelerated to about 13 times this energy as the plasma expands a factor of 10 in radius, and the density drops by a factor of about 360. The acceleration is caused by the ambipolar electric field generated by electron thermal pressure.

The steady-state radial profiles of ion density and average ion energy obtained from Aleph simulations agree with the analytic model. The numerical heating of the ions can be reduced at the expense of increased computation time by reducing the timestep and by increasing the number of macroparticles. The fact that temporal smoothing was found to have little effect on the electric field noise suggests that the noise is spatial rather than temporal. Spatial smoothing similar to that associated with a Poisson solver may be beneficial.

The ambipolar model assumes a constant, uniform electron temperature. A previous, fully-kinetic plasma-expansion calculation with Aleph (see Fig. 8 in Ref. [4]) showed evidence that electron temperature decreases as a function of radius. (The plasma expanded just a factor of two in radius.) This effect could be captured in Aleph by using a pressure-density relation of the form $p/n^\gamma = \text{constant}$, where γ is the "adiabatic index". For the isothermal model used in the present calculations, $\gamma = 1$, while for adiabatic expansion of an ideal gas, $\gamma = 5/3$. The value $\gamma \approx 4/3$ gave a reasonably good fit to the kinetic electron temperature in Ref. [4].

It may be worthwhile to investigate a compressible-fluid treatment based on the Euler equations for both ions and electrons in order to avoid the numerical heating seen in the calculations described here.

References

[1] R. J. Mason. Implicit moment particle simulation of plasmas. *J. Comp. Phys.*, 41:233, 1981.

[2] B. I. Cohen, A. B. Langdon, and A. Friedman. Implicit Time Integration for Plasma Simulation. *J. Comp. Phys.*, 46:15, 1982.

[3] G. Chen, L. Chacón, and D. C. Barnes. An energy- and charge-conserving, implicit, electrostatic particle-in-cell algorithm. *J. Comp. Phys.*, 230:7018, 2011.

[4] T. P. Hughes and E. V. Barnat. Current flow to a biased electrode in a vacuum-arc plasma. Technical report SAND2011-4343, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, June 2011. (link to EIMS copy).

A Cubit input file

The mesh in Fig. 2 was generated with Cubit 14.1. The following mesh information was written by Cubit:

```
Executive Exodus summary:
  Number of dimensions     = 3
  Number of element blocks = 1
  Number of sidesets       = 3
  Number of nodesets       = 3
  Number of bc sets        = 1
  Number of elements       = 201467
  Number of nodes          = 37402
```

The Cubit input file is listed below:

```
## /projects/cubit/claro.Lin64.14.1/bin/clarox
## Cubit Version 14.1
## Cubit Build 390213
## Revised 2014-01-20 08:40:22 -0700 (Man, 20 Jan 2014)
## Running 01/24/2014 10:20:45 AM
## Command Options:
## -warning = On
## -information = On

# spherical half-wedge (20 degrees)

# Parameters
# {rmin=10.0}
# {rmax=100.0}
# {Lmax=2*rmax+1.0}
# {half_angle=10.0}
# {dx=1.745}
#
reset
undo on

# Create the inner and outer spheres
Create sphere radius {rmax}
create sphere radius {rmin}
volume 1 rename "outer_sphere"
volume 2 rename "inner_sphere"
chop outer_sphere with inner_sphere
delete inner_sphere

#Trim the sphere into a frustum with bricks
create brick x {Lmax} y {Lmax} z {Lmax}
move volume 5 x {Lmax/2}

rotate volume 5 about y angle {half_angle}

create brick x {Lmax} y {Lmax} z {Lmax}
move volume 6 x {-Lmax/2}
rotate volume 6 about y angle {-half_angle}
```

```
subtract volume 5 from volume 4
subtract volume 6 from volume 4

# Cut the slice in half
create brick x {Lmax} y {Lmax} z {Lmax}
move volume 7 y {Lmax/2}
subtract volume 7 from volume 4

# Generate sidesets and nodesets
sideset 1 surface 31
nodeset 1 surface 31
sideset 2 surface 33
nodeset 2 surface 33
sideset 3 surface 29 30 32
nodeset 3 surface 29 30 32

# Create the tet mesh
volume all size {dx}
volume all scheme tetmesh
mesh volume all

block 1 volume all
block all element type tetra4

# Change units to meters for Aleph
transform mesh output reset
transform mesh output scale 1.0e-3

export genesis "slice3.g" block all dimension 3 overwrite
```

B Aleph input file

Aleph version 4454 was run using the commands:

```
aprepro -d expansionH4.in expansionH4A.in
./aleph expansionH4A.in >&alog4&
```

The following is the input file for the reference calculation described in Sec. 4.

```
Sensitivity Level = UUR

# Model an expanding quasineutral plasma

# new mesh: 1/2-slice of a 20-degree spherical edge
# new mesh: half the resolution

# Compile flags:
# -j8 -DBUILD_H5PART=1

### APREPRO Section

## CONSTANTS

# constants
# {nsec = 1.0e-9} seconds
# {eV = 11604.0} K
# unit charge {qe = 1.6e-19} Coulomb
# atomic mass unit {AMU = 1.66e-27} kg
# Boltmann constant {kB = 1.38E-23} J/K

## PHYSICS SETUP

# electron temperature (eV) {Te_eV = 1.0}
# electron temperature (K) {Te_K = Te_eV*eV}
# ion charge state {ZI = 1.0}
# ion charge (C) {qI_C = ZI*qe}
# ion mass (AMU) {mI = 1.0}
# ion mass (kg) {mI_kg = mI*AMU}
# Bohm velocity {vB = sqrt(ZI*kB*Te_K/mI_kg)}

# ion density at injection {n_inj = 2.0e17} per m3
# ion speed at injection {v_inj = vB} m/s

# ion weighting {wI = 1.25e6}

## TIMESTEPS

# timestep {dt = 60.0*nsec}
# time limit {tmax = 18000.0*nsec}
# timesteps {nsteps = nint(tmax/dt)}

# rebalance every {nbal = 100} steps

## OUTPUT

## sliding window size for ambipolar field (steps) {win_size = 100}
```

```
## desired number of frames in field output {fframes = 20}
# Exodus stride {fstride = int(nsteps/fframes)}
#               {fstride = max(fstride, 1)}
## number of timesteps to average fields over {n_avg = 100}
# Take the minimum: {n_avg = min(n_avg, fstride)}

## desired number of frames in particle output {pframes = 20}
# H5PART stride {pstride = int(nsteps/pframes)}
#               {pstride = max(pstride, 1)}

## desired number of samples in probe output {hsamp = 200}
# probes stride {hstride = int(nsteps/hsamp)}
#               {hstride = max(hstride, 1)}

### END APREPRO Section


# General controls

units = SI
random number generator seed = 314

# time

timestep size = {dt} # seconds
total number of timesteps = {nsteps}

# restarts
restart stride = off
restart file name = restart1.dat, toggle=restart2.dat

#read restart file = restart2.dat

# field solve
electromagnetic model for block_1 = ambipolar, Te = {Te_K}, \
                output_name = E_ambi \
                ion_density_window = [sliding, size={win_size}], \
                ion_density_smoothing_level  = 0, \
                ion_density_output_name = n_ambi, \
                ion_gradient_smoothing_level  = 0, \
                ion_gradient_output_name = dn_ambi

potential field solve stride = off

# parallelization
rebalance stride = {nbal}, method = geometric, weighting = constant, \
          schedule = constant
output load rebalancing particle data = yes

# mesh
#Executive Exodus summary:
#  Number of dimensions     = 3
#  Number of element blocks = 1
#  Number of sidesets       = 3
#  Number of nodesets       = 3
#  Number of bc sets        = 1
#  Number of elements       = 201467
#  Number of nodes          = 37402
#Finished writing slice3.g

input mesh file name = slice3.g

#
# Particle types
```

```
#

define particle H+ mass={mI_kg}, charge={qI_C}, category=ion

#
# Initial Conditions
#

# particles

particle weighting for H+ =  {wI}

#
# Boundary conditions
#

# set 1: outer radius
#     2: inner radius
#     3: sidewalls

# fields
# no field BCs for ambipolar

# particles

BC for particles on default is specular for default

# inject ions
BC for particles on surface_2 is normal_influx of H+ at density={n_inj}, \
   T = 0, vnormal = {v_inj}

# outflow at injection surface
BC for particles on surface_2 is outflux for default

# outflow at outer surface
BC for particles on surface_1 is outflux for default

#
# Interactions
#
interaction stride = off

#
#  Reweighting
#
reweighting stride = off

#
# Output
#

# fields on nodes/elements

Exodus output file name = f.ex2
Exodus output stride = {fstride}

# number densities
output nodal density for H+, name = n_Hp_nd, element_block=all, \
       window=discrete,size=1
output nodal density for H+, name = n_Hp_nd_avg, element_block=all, \
       window=discrete,size={n_avg}

# summed charge density
output solver charge density = true, name = rho, element_block=all, \
       window=discrete,size={n_avg}
```

```
# injected current
output nodal current_density for H+ on surface_2, name= j_Hp_inj, \
       window=discrete,size=1

# outflow current
output nodal current_density for H+ on surface_1, name = j_Hp_out, \
       window=discrete,size=1

# temperatures
output elemental temperature for H+, name = temp_Hp_el, \
       window=discrete,size=1

# electric field (automatic)

# diagnostics
output elemental computational_count for H+, name = Hp_count_el, \
       window=discrete,size=1
output elemental proc_ids = true
output elemental global_ids, element_block = all, name = gid_el

# particles

#particle dump file name = particles.lps
particle dump file name = p.h5part, file type=h5part, attributes = local_id, \
         type, x, y, z, vx, vy, vz, stride={pstride}, fraction = 1

# status reports

output global computational_count for H+, name =        H+_count, \
       window=discrete,size={hstride},stride={hstride}

# injection
output global current for H+ on surface_2, name = H+_current_inj, \
       window=discrete,size={hstride},stride={hstride}
output global integrated_flux on surface_2 for H+, name = H+_flux_inj, \
       window=discrete,size={hstride},stride={hstride}
output global particle_kinetic_energy on surface_2 for H+, name=H+_KE_inj, \
       window=discrete,size={hstride},stride={hstride}

# outflow
output global current for H+ on surface_1, name =       H+_current_out, \
       window=discrete,size={hstride},stride={hstride}
output global integrated_flux on surface_1 for H+, name=H+_flux_out, \
       window=discrete,size={hstride},stride={hstride}
output global particle_kinetic_energy on surface_1 for H+, name=H+_KE_out, \
       window=discrete,size={hstride},stride={hstride}

# status output

status report name = stdout columns = timestep, simulation_time, H+_count, \
       stride={hstride}

status report name = globals.dat format = gnuplot,\
 columns= timestep,\
 simulation_time,\
 H+_count,\
 H+_current_inj,\
 H+_flux_inj, \
 H+_KE_inj, \
 H+_current_out,\
```

```
240     H+_flux_out, \
        H+_KE_out, \
        stride={hstride}

      status report name = memory.dat format = gnuplot,\
        columns = timestep,\
        simulation_time,\
        high_mb_min_used,\
        high_mb_max_used,\
        high_mb_min_free,\
250     high_mb_max_free,\
        stride={hstride}

      status report name = seq_timers.txt columns = sequential_timers stride={hstride}
```

DISTRIBUTION:

1	MS 0822	W. Scott, 9326
1	MS 0828	A. Geller, 1516
1	MS 0836	J. Boerner, 1516
1	MS 0840	D. Rader, 1513
1	MS 0862	C. Busick, 2734
1	MS 0862	R. Ferrizz, 2734
1	MS 0862	K. Meredith, 2734
1	MS 0869	C. Renschler, 2730
1	MS 0878	S. Dirk, 2735
1	MS 0878	J. Howard, 2735
1	MS 0878	T. Hughes, 2735
1	MS 0878	G. Laity, 2735
1	MS 0878	M. Lovejoy, 2735
1	MS 1168	M. Bettencourt, 1352
1	MS 1168	K. Cartwright, 1352
1	MS 1168	C. Moore, 1352
1	MS 1177	L. Musson, 1355
1	MS 1318	S. Moore, 1444
1	MS 1318	R. Hooper, 1446
1	MS 1322	P. Crozier, 1444
1	MS 1423	E. Barnat, 1118
1	MS 1423	P. Miller, 1118
1	MS 0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories